REACT: Revealing Evolutionary Action Consequence Trajectories for Interpretable Reinforcement Learning.

Philipp Altmann, Céline Davignon, Maximilian Zorn, Fabian Ritz,
Claudia Linnhoff-Popien and Thomas Gabor
LMU Munich, Germany

Keywords: Reinforcement Learning, Interpretability, Genetic Algorithms.

Abstract: To enhance the interpretability of Reinforcement Learning (RL), we propose Revealing Evolutionary Action Consequence Trajectories (REACT). In contrast to the prevalent practice of validating RL models based on their optimal behavior learned during training, we posit that considering a range of edge-case trajectories provides a more comprehensive understanding of their inherent behavior. To induce such scenarios, we introduce a disturbance to the initial state, optimizing it through an evolutionary algorithm to generate a diverse population of demonstrations. To evaluate the fitness of trajectories, REACT incorporates a joint fitness function that encourages local and global diversity in the encountered states and chosen actions. Through assessments with policies trained for varying durations in discrete and continuous environments, we demonstrate the descriptive power of REACT. Our results highlight its effectiveness in revealing nuanced aspects of RL models' behavior beyond optimal performance, with up to 400% increased fidelities, contributing to improved interpretability. Code and videos are available at `https://github.com/philippaltmann/REACT`.

1 INTRODUCTION

With the increasing use of large, parameterized function approximation models, there is a growing demand for interpretation methods that bridge the gap between human understanding and computational intelligence. This is particularly pronounced in the context of complex dynamic approaches like reinforcement learning (RL), where policies are usually realized with parameterized neural networks. As a running example, consider a 9×9 gridworld, where the agent is perfectly trained to traverse the environment and reach the target field. However, unforeseen circumstances (like sensor failure or domain shifts) might cause the agent to end up in fields not along this optimal trajectory, where an overfitted policy might even get stuck. Yet, those scenarios are equally important to interpret the inherent behavior. This yields several challenges: First, contrary to static supervised learning tasks like classification, RL policies are inherently hard to visualize, especially given the intended application to varying circumstances. Second, demonstrating the desired behavior in a laboratory training setup does not serve as sufficient validation to enable the interpretability of the inherent behavior. Third, *comparative evaluation* plays a central role in comprehending, explaining, and interpreting varying phenomena by providing additional context information and, thus, control (Vartiainen, 2002). To tackle these challenges, we propose to evaluate a set of diverse edge-case demonstrations, which we obtain by precisely disturbing the initial state. To generate a small yet informative set of demonstrations, we employ evolutionary optimization, which can be adapted to yield diverse solution candidates in complex solution landscapes across various (local) optima. To harness these prospects, we propose a framework to indirectly optimize a population of demonstration behavior generated by a given (trained) policy by altering (disturbing) the initial state. Overall, we provide the following contributions:

- We formalize a novel interpretability joint fitness metric to assess demonstration trajectories w.r.t. their local (inherent) and global (comparative) state diversity and action certainty.
- We propose an architecture for *Revealing Evolutionary Action Consequence Trajectories* (REACT), integrating the previously defined fitness to optimize a pool of diverse demonstrations to serve as a basis for interpreting the underlying policy.
- We evaluate REACT in flat and holey gridworlds and a continuous robotic control task, comparing policies of varying training stages.

Altmann, P., Davignon, C., Zorn, M., Ritz, F., Linnhoff-Popien, C. and Gabor, T.
REACT: Revealing Evolutionary Action Consequence Trajectories for Interpretable Reinforcement Learning.
DOI: 10.5220/0013005900003837

In *Proceedings of the 16th International Joint Conference on Computational Intelligence (IJCCI 2024)*, pages 127-138
ISBN: 978-989-758-721-4; ISSN: 2184-3236

2 PRELIMINARIES

Reinforcement Learning. We focus on problems formalized as *Markov decision processes* (MDPs) $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mu, \gamma \rangle$, with a set $\mathcal{S}$ of states s, a set $\mathcal{A}$ of actions a, a transition probability $\mathcal{P}(s' \mid s, a)$ of reaching s' when executing a in s, a scalar reward $r_t = \mathcal{R}(s, a, s') \in \mathbb{R}$ at step t, the initial state distribution $s_0 \sim \mu$, and the discount factor $\gamma \in [0, 1)$ for calculating the discounted return $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ (Puterman, 1990). More specifically, we consider learning in a constrained setting with a single deterministic initial state $s_0^\star$ and evaluating with initial states drawn from μ. Furthermore, we consider the objective of *reinforcement learning* (RL) to find an optimal policy π^* with action selection probability $\pi(a \mid s)$ that maximizes the expected discounted return (Richard S. Sutton, 2015). *Policy-based* methods directly approximate the optimal policy from trajectories τ of experience tuples $\langle s, a, r, s' \rangle$, generated by π. *Proximal policy optimization* (PPO) extends this concept, optimizing a surrogate loss that restricts policy updates to improve the robustness (Schulman et al., 2017). *Soft actor-critic* (SAC) bridges the gap between value-based and policy-based approaches (Haarnoja et al., 2018). Both algorithms have shown versatile applicability to various scenarios. Thus, we use both approaches to train the policies we base our empirical studies on. While enabling learning in complex high-dimensional or continuous scenarios, using deep neural networks to approximate the optimal policy comes at the cost of introducing a black-box model. Therefore, even when finding a parameterization that resembles an optimal policy, its decision cannot be anticipated, and reasons for action choices cannot be (readily) inferred. Yet, RL has been proposed to provide compelling solutions to various real-world decision-making problems such as autonomous driving or robotic control (Wurman et al., 2022; de Lazcano et al., 2023; Rolf et al., 2023). Such problems require transparency, e.g., to account for safety concerns or quality control.

Explainability. This field of research not only concerns providing explanations for specific decisions of such black-box models but also extends to providing their general interpretability. According to Li et al. (2022), we classify interpretation algorithms regarding three characteristics: Their representation, the type of the model to be interpreted, and the relation between the interpretation algorithm and the model. The representation can be based on the *importance* of (latent) features in relation to the final objective (Lundberg and Lee, 2017). Alternatively, one can use the *model's response* to different inputs to identify behaviors. Some algorithms approximate the model using an interpretable surrogate model (Ribeiro et al., 2016). Finally, some models show the interpretation by a *sample dataset* showing the impact of training (Koh and Liang, 2017; Pleiss et al., 2020). Regarding the model to be interpreted, some approaches consider the model as a black box (Pleiss et al., 2020; Ribeiro et al., 2016). These algorithms are called *model-agnostic* and can be applied to any model. Other approaches require specific model characteristics such as differentiability or even a particular type of model (Koh and Liang, 2017). *Closed-form* algorithms are applied after training, while *composition* algorithms can (also) be integrated into the training process. Further relations include *dependence*, where the algorithms add operations to the model after training to output interpretable terms, and *proxy*, where an interpretable proxy model is created. Our algorithm represents the interpretation as a *model response*, displaying the policy behavior throughout various trajectories provoked by the initial state. Furthermore, we consider the model a black box, where our algorithm can interpret various models, provided any action selection probability. The type of model is not relevant to our approach, making our approach model-agnostic. Furthermore, we propose a closed-form approach to be applied after training.

Evolutionary Optimization. To optimize initial states that cause diverse demonstrations, we use a population-based evolutionary optimization process with populations $\mathbb{P} = \{\tau_i\}_{0 \leq i \leq p}$ of size p, where the initial population $\mathbb{P}_0$ is chosen randomly, state space $\mathbb{X}$ with $\mathbb{P} \in \mathbb{N}^{\mathbb{X}}$, a fitness function $\mathcal{F} : \mathbb{X} \to \mathbb{R}$, and the evolution step function $E(\mathbb{P}_t, \mathcal{F}) = \mathbb{P}_{t+1} = \sigma_p\big(\mathbb{P}_t \uplus \texttt{mutants}_{p_m}(\mathbb{P}_t) \uplus \texttt{children}_{p_c}(\mathbb{P}_t)\big)$, with a (non-deterministic) selection function $\sigma_n : \mathbb{N}^{\mathbb{X}} \to \mathbb{N}^{\mathbb{X}}$ that returns $n \in \mathbb{N}$ individuals and could depend on $\mathcal{F}$, a mutation function $\texttt{mutants}_{p_m} = \{\texttt{mutation}(x) : x \sim \sigma_{\lceil p \cdot p_m \rceil}\}$ and a crossover function $\texttt{children}_{p_c} = \{\texttt{crossover}(x_1, x_2) : x_1, x_2 \sim \sigma_{\lceil p \cdot p_c \rceil}\}$, with mutation and crossover probabilities (rates) p_m and p_c (Fogel, 2006). Individuals $\mathcal{I} \in \mathbb{P}$ are defined by their inherent features (*genotype*), in which we encode an initial state s_0 sampled from the initial state distribution μ defined by the MDP. Their individual fitness is calculated based on their resulting appearance (*phenotype*), i.e., the demonstration trajectory τ generated by executing policy π in the given environment starting from s_0. A binary encoding of the individual state allows for implementing a simple *bit-flip* mutation a *single-point crossover* operation to recombine two parents.

To foster parents with higher fitness, *tournament selection* is commonly applied within function σ (Miller et al., 1995). While evolutionary algorithms are usually used to search for one single best individual, we are interested in the entire population of individuals similar to (Ishibuchi et al., 2008; Neumann et al., 2019). Deploying a fitness function that promotes diversity among trajectories allows us to see the different strategies an agent follows in different situations. Generally, all measures of the diversity of an individual $\mathcal{I}$ in a population $\mathbb{P}$ are related to the pairwise distance between individuals in $\mathbb{P}$ as measured by a suitable norm (e.g., Euclidean for real-valued representations, Hamming for symbolic representations) (Wineberg and Oppacher, 2003). Therefore, the individual diversity w.r.t. the population can be estimated by $\mathcal{D}(\mathcal{I}, \mathbb{P}) = \frac{1}{p} \cdot \sum_{\mathcal{I}' \in \mathbb{P}} |\mathcal{I}' - \mathcal{I}|$ (Gabor et al., 2018).

3 TRAJECTORY FITNESS EVALUATION

In the following, we discuss assessing the fitness of trajectories $\tau = \langle s_0, a_0, r_1, \ldots, s_t, a_t, r_{t+1} \rangle \sim \mathcal{P}_{\pi, s_0}$ to serve as an *insightful* demonstration to interpret the inherent behavior of policy π. Unlike the central objective of RL, we are not interested in optimizing for the best-performing individuals but rather in a population of diverse demonstrations following π from an initial state s_0 to be optimized. Therefore, we refrain from using the reward metric supplied to learn the policy and define a joint fitness metric $\mathcal{F}$ in the following. To illustrate our deliberations, we consider the 9×9 gridworld environment depicted in Fig. 1.

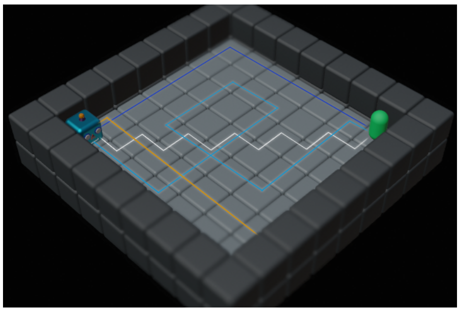


Figure 1: Joint fitness $\mathcal{F}$ elements *local diversity* $\mathcal{D}_l$ (light blue), *global diversity* $\mathcal{D}_g$ (blue), and *certainty* $\mathcal{C}$ (orange), compared to an exemplary optimal trajectory (white).

We strive for high diversity to achieve *insightful* demonstrations. Considering a single trajectory, a diverse path covering a larger fraction of the available state space (e.g., the light blue path in Fig. 1) would be more informative regarding the behavior to be analyzed than the comparably direct path resulting from policy optimization (e.g., the white path in Fig. 1). Even though it might be considered less optimal w.r.t. the reward of the given environment, such behavior might depict an edge, which is important to assess the given policy. We refer to this measure as *local diversity* and formalize the corresponding metric

$$\mathcal{D}_l(\tau) = \frac{1}{|P|} |\{s \in \tau\}|, \tag{1}$$

where $P = \{P_d \mid P_d \subset \mathbb{N}, \forall d \in 1, \ldots, dim\}$, with $|P| = |P_1| \cdot \ldots \cdot |P_{dim}|$ is the dim-dimensional position space extracted by $\rho : \mathcal{S} \mapsto P$ from a state s. In our exemplary gridworld, we consider the 2-dimensional position of the agent with a $|P| = 9 \cdot 9$ distinct states. Yet, this representation might be extended by other important, moving, or task-specific objects like obstacles or targets. In our case, higher local diversity implies more divergence from the optimal path, increasing the relevance of the trajectory. Furthermore, this position-centric formalization allows us to consider the Euclidean distance between states $||s - s'||_2 = \sqrt{(\rho(s) - \rho(s')^2}$. For use in continuous environments, we suggest applying appropriate discretizations to regularize state similarities.

Considering a set of multiple trajectories $\mathcal{T}$, neither solely disturbed (light blue) nor solely optimal (white) paths accurately reflect the behavior of π. We therefore additionally consider a *global diversity* $\mathcal{D}_g$ (blue) of trajectories $\tau \in \mathcal{T}$ formalized as

$$\mathcal{D}_g(\tau, \mathcal{T}) = \frac{1}{\lceil P \rceil} \min_{\tau' \in \mathcal{T} \setminus \tau} \delta(\tau, \tau'), \tag{2}$$

based on the *maximum state distance* $\lceil P \rceil = \max_{s, s' \in \mathcal{S}} ||s - s'||_2$ and the *one-way distance* δ between trajectories τ and τ' (Lin and Su, 2008):

$$\delta(\tau, \tau') = \frac{\sum_{s \in \tau} d(s, \tau') + \sum_{s' \in \tau'} d(s', \tau)}{|\tau| + |\tau'|}, \tag{3}$$

using the state-to-trajectory distance:

$$d(s', \tau) = \min_{s \in \tau}(||s - s'||_2) \tag{4}$$

This accumulated two-way measure allows for comparison between trajectories of different lengths. Furthermore, using the min operator in Eq. (2) causes equal trajectories in $\mathcal{T}$ to be valued at 0. Ultimately, even if $\mathcal{T}$ contains only optimal yet maximally dissected behavior to reach the target, presenting such diverse demonstrations increases the overall interpretability of π. Note that, even though only defined for disturbing the agent's position, further deviations, such as altering layouts, are formally not precluded. However, calculating the global diversity might require using a different distance metric, like the Levenshtein distance, instead.

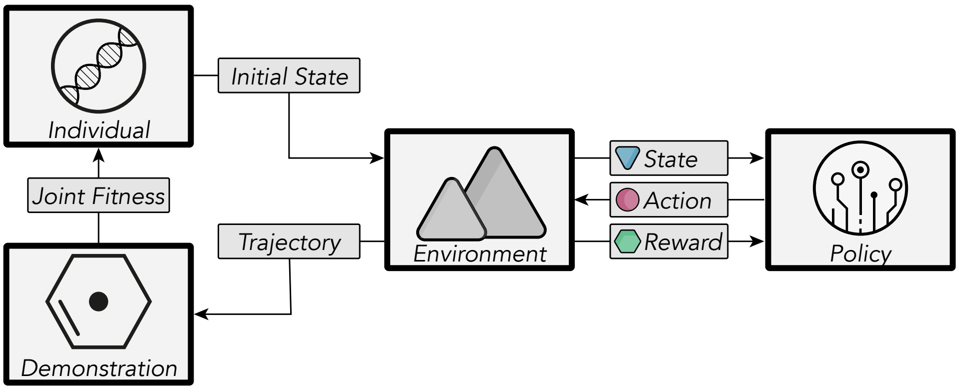


Figure 2: REACT Architecture.

Both diversity measures implicitly cover insufficiencies and uncertainties of π that may occur in states less prevalent during training. To reflect the diversity of the action decision itself, we furthermore consider the *certainty*, formalized as the cumulative normalized action probability of τ given π:

$$\mathcal{C}_\pi(\tau) = \frac{1}{|\tau|} \sum_{s,a \in \tau} \pi(a|s). \tag{5}$$

Counterintuitively, we are interested in trajectories with low certainties causing more diverse decisions that may fail to solve the intended task, such as the exemplary orange path in Fig. 1. We intentionally chose the normalized sum of probabilities instead of their product to promote trajectories with low certainties throughout.

Overall, we define the *joint fitness*, combining *global diversity* $\mathcal{D}_g$, *local diversity* $\mathcal{D}_l$, and *certainty* C_π of a trajectory τ in context of a set of previously evaluated trajectories $\mathcal{T}$ as follows:

$$\mathcal{F}(\tau, \mathcal{T}) = \mathcal{D}_g(\tau, \mathcal{T}) + \mathcal{F}_l, \text{ with} \tag{6}$$

$$\mathcal{F}_l = \min_{t \in \mathcal{T}} \left\| \begin{pmatrix} \mathcal{D}_l(\tau) \\ \mathcal{C}_\pi(\tau) \end{pmatrix} - \begin{pmatrix} \mathcal{D}_l(t) \\ \mathcal{C}_\pi(t) \end{pmatrix} \right\|_2 . \tag{7}$$

To reflect the τ-specific metrics of local diversity and certainty in relation to the set of trajectories $\mathcal{T}$, considered for calculating the *global diversity*, we consult these measures only regarding their minimum distance between τ and $\mathcal{T}$. We chose the minimum distance of both local metrics to encourage individuals to maximize their local distance to the closest individual, thereby promoting diverse or uncertain behavior. As we defined all components of the joint fitness to be normalized, we furthermore do not introduce additional parameterizations to balance their impact. Preliminary studies confirmed this approach.

4 REACT

To optimize a pool of demonstrations to interpret a given policy using the previously defined fitness, we propose *revealing evolutionary action consequence trajectories* (REACT) to optimize a population of initial states causing diverse demonstrations. By showing not only the optimistic optimal behavior, we aim to increase the traceability of the learned behavior and, ultimately, trust in the black-box policy model. In contrast to most evolutionary approaches, we are interested in the whole population, not just the single best-performing individual. The overall architecture is depicted in Fig. 2 and outlined in Alg. 1.

To form the initial population $\mathbb{P}$ of size p, individuals encoded by the initial state s_0 are generated from μ given by the MDP of the given environment. Invalid individuals that cannot generate any demonstration are disregarded. As we only introduce disturbances to the initial position of the agent, the initial state s_0 can be encoded by the initial position of the agent. To account for evaluation environments comprising different-sized 2D-discrete and 3D-continuous state spaces, we opt for a universal multi-dimensional 6-bit encoding with inverse normalization to ensure precise reconstruction of the intended position. For further details, please refer to the appendix.

To evaluate the individuals' fitness, trajectories τ are sampled from the environment, starting from their individual initial state, following π. For improved comparability, we furthermore remove duplicate consecutive states from τ. These demonstrations constitute the individuals' phenotype directly affecting their fitness to serve as a viable representation of the given model. The individual fitness is calculated according to Eq. (6) based on the individual trajectory τ and the set of previous demonstrations $\mathcal{T}$ to reflect the individual performance within the demonstration pool. Note that even though we sample experiences from

Algorithm 1: Revealing Evolutionary Action Consequence Trajectories (REACT)[1].

```
Require: $\mathcal{P}, \mu, \pi$                                  ▷ We use a policy trained with a single initial state
$\mathbb{P} \leftarrow \langle s_0 \sim \mu \rangle_p$ ; $\mathcal{T} \leftarrow \emptyset$      ▷ Generate initial population of size $p$ and empty $\mathcal{T}$
for individual $\mathcal{I} \in \mathbb{P}$ do
    $\tau \sim \mathcal{P}_{\pi,s_0}$                              ▷ Sample trajectory $\tau_\mathcal{I}$ from initial state $s_0$
    $\mathcal{F}_\mathcal{I}(\tau, \mathcal{T})$                  ▷ Calculate Fitness of $\mathcal{I}$ w.r.t. to phenotype $\tau$ and
                                                                    previous demonstrations $\mathcal{T}$ according to Eq. (6)
    $\mathcal{T} \leftarrow \mathcal{T} \cup \tau$                ▷ Update demonstrations $\mathcal{T}$
end for
for all generations $g$ do
    $\mathbb{O} \leftarrow \mathtt{mutants}_{p_m}(\mathbb{P}) \uplus \mathtt{children}_{p_c,\mathcal{F}}(\mathbb{P})$  ▷ Generate offspring from mutation and crossover using
                                                                    $p_m$, $p_c$, the individual fitness, and tournament selection
    for individual $\mathcal{I} \in \mathbb{O}$ do
        $\tau \sim \mathcal{P}_{\pi,s_0}$                          ▷ Sample trajectory $\tau_\mathcal{I}$ from initial state $s_0$
        $\mathcal{F}_\mathcal{I}(\tau, \mathcal{T})$              ▷ Calculate Fitness according to Eq. (6)
        $\mathcal{T} \leftarrow \mathcal{T} \cup \tau$            ▷ Update demonstrations $\mathcal{T}$
    end for
    $\mathbb{P} \leftarrow \mathtt{migration}(\mathbb{P} \uplus \mathbb{O}, \mathcal{F}, p)$   ▷ Select $p$ best individuals for the next generation from
                                                                    the population and offspring according to their fitness
    $\mathcal{T} \leftarrow \mathcal{T} \setminus \{\tau_\mathcal{I} \mid \mathcal{I} \notin \mathbb{P}\}$   ▷ Remove extinct demonstrations
end for
return $\mathcal{T}$
```

the environment, we do not consider further improving the policy at hand. Nevertheless, the proposed architecture could serve as an automated adversarial curriculum to generate scenarios for further training.

After evaluating the first generation, the best individuals are selected via tournament selection to create new individuals through recombination. The recombination operator is executed with the recombination probability $p_c \in [0, 1]$ defined beforehand. To generate the offspring, we use single-point crossover. The new individuals are then added to the population. Then, a mutation operator with mutation probability $p_m \in [0, 1]$ is applied to random individuals from the original population. The mutation is implemented by a single bit-flip of one random bit in the individual's encoding. As we are interested in the whole population, we keep the individual before mutation and add the mutated individual to the population to keep the evolution elitist. After evaluating the newly generated offspring, as described above, one after the other, the population is reduced to the intended size p by removing the individuals with the lowest fitness value along with their generated demonstrations. The described procedure is repeated for a fixed number of g generations.

Hyperparameters. The most important hyperparameter to consider is the population size p. It influences the effectiveness of the evolutionary process and determines the number of demonstrations generated to interpret the policy. To suit human needs, p should be comprehensibly small and sufficiently diverse (Behrens et al., 2023). Preliminary experiments suggest a population size of $p = 10$ is a reasonable compromise. Larger populations can be used if only the best p individuals are considered to demonstrate the policy's behavior. For experimental details, please refer to the appendix[1]. Furthermore, if not stated otherwise, we optimize the population of demonstrations over 40 iterations (generations). Our central goal is to diversify the population throughout optimization, so we use a reasonably high *crossover probability* $p_c = 0.75$ combined with a high *mutation probability* $p_m = 0.5$. In combination with the chosen binary state encoding of length 6, representing the agent's initial position, this configuration causes the generation of offspring to start at further distances.

5 RELATED WORK

Evolutionary RL. Evolutionary approaches have also been applied to optimize a population of policies (Khadka and Tumer, 2018) to foster their explorative capabilities. Both task-agnostic (Parker-Holder et al., 2020) and task-specific (Wu et al., 2023) diversity

[1] All required implementations, appendices, and video renderings are available at https://github.com/philippaltmann/REACT.

measures have been shown to be beneficial for improving the quality of the resulting policy. Note, however, that we do not consider any policy improvement but use evolutionary optimization to generate scenarios that best describe the learned policy in a given environment. Nevertheless, this line of work highlights the importance of considering the diversity of behavior in addition to its quality. Similarly, *quality diversity* (QD) optimization arose from considering the behavioral novelty of solution candidates as their optimization criteria (Lehman and Stanley, 2011). Gabor and Altmann (2019) proposed using surrogate-assisted genetic algorithms for building recommender systems. Bhatt et al. (2022) integrated QD into the automated environment generation during training via a surrogate model to improve the robustness of the policy. We take a similar approach to improve the interpretability of the learned policy, optimizing for a set of diverse policy demonstrations. However, in contrast to the novelty criterion, which considers solely the distance to the current population, we propose using a joint fitness combining both local and global criteria of the trajectories to be optimized.

Robust RL. We consider a process where a policy is trained with a single deterministic initial state and evaluated with a changing initial state to simulate the policy's edge-case behavior, allowing the learned behavior to be interpretable. Therefore, from a different perspective, we consider the robustness of a policy to out-of-distribution samples, i.e., initial states that were potentially not experienced during training, also referred to as generalization capabilities. If an agent is trained well, only looking at some episodes of the agent's interaction with the environment usually solely shows the expected behavior, including often-occurring states. However, the agent's strategy also includes behavior in states that have not been encountered that often. We also want to show this behavior. The goal is to show the most diverse behavior and generate a small but informant overview of the agent's strategy. To improve the generalization capabilities, using varying training configurations (Cobbe et al., 2020), optimized training scenarios (Altmann et al., 2023), or an evolving curriculum (Parker-Holder et al., 2022) has shown to be a viable approach. Yet, we specifically chose a different training approach to showcase the methodical impact of REACT for visualizing a possibly insufficient policy in edge-case scenarios. Note that this work generally does not consider any policy improvement. Nevertheless, the generated representations could be fed back into the training process as *adversarial samples*, similar to Gabor et al. (2019).

Explainable RL. There are several approaches to the *interpretability and explainability of RL* (XRL), which are surveyed by Heuillet et al. (2021) and Alharin et al. (2020). Similar to general explainability approaches previously introduced, RL interpretation algorithms can be divided into different categories. One central aspect is their scope, reflecting either local decisions or the global strategy. A further distinction is drawn between post-hoc methods, which keep the original model (Lage et al., 2019), and intrinsic methods, replacing the original model with a more explainable surrogate (Guo et al., 2021; Huang et al., 2017). Combinations of both are also possible. Furthermore, XRL algorithms can be applied before, during, or after training. Finally, XRL algorithms can be classified according to their type of explanation. The most common types are textual explanations, image explanations, collections of states or state-action pairs, and explanations through rules. We approach XRL by generating a **collection of demonstration trajectories** that show diverse behavior based on a given policy interacting with an environment. We thereby strive for a scope that includes the **global inherent strategy**. Furthermore, we optimize the diversity of those demonstrations using an evolutionary process, which can be considered a **post-hoc method**. Specifically, REACT does not require a particular policy specification and, therefore, does not need to be integrated prior to or during training. Similarly, Amir and Amir (2018) propose creating a policy summary containing a fixed number of important states and their surrounding states. The effect of an action on that state identifies the importance of states. The goal is to find states where a slight action modification would strongly influence the cumulative reward. Therefore, the approach is mainly based on the value function rather than relying on an external optimization mechanism. Such states have also been referred to as *critical states*, where the chosen action has a significant impact on the outcome, which can be used to interpret policies trained using maximum entropy-based RL (Huang et al., 2018). While REACT is similar regarding its global scope, we refrain from integrating the reward into the fitness to be optimized and instead use it to validate finding a set of diverse demonstrations. Likewise, Sequeira and Gervasio (2020) consider the agents' actions and states in the environment, but also its policy, to compile a summarizing video of *interestingness elements*. The frequency, execution certainty, transition value, and sequences determine interesting elements, intending to show a maximally diverse set of highlights. In contrast to both, we consider demonstrations of complete trajectories instead of patching together possibly unrelated sequences due to their impact. Therefore, we refrain from experimental comparison to those approaches. Nevertheless, to measure the quality of demonstrations, we use the fidelity metric proposed by Guo et al. (2021), adapted to indicate increased fidelity with higher scores, which we introduce in the following section.

RL Testing. Like REACT, Tappler et al. (2022) use a genetic algorithm to find an *interesting* trace for testing RL policies. Zolfagharian et al. (2023) optimize full episodes to search for faulty behavior and train a predictive model. REACT, on the other hand, optimizes the initial state that causes the trace, given a policy to be analyzed. Pang et al. (2022) propose a similar fuzz test framework for RL, modifying the initial state to generate *fresh* sequences. In contrast to REACT, they do so by estimating the sensitivity of the given model to its seed instead of applying evolutionary operators on the initial state. Overall, however, those approaches are primarily motivated to generate test cases, preferably where the model fails. REACT aims to generate a balanced representation of the learned behavior, specifically including edge cases.

6 EVALUATION

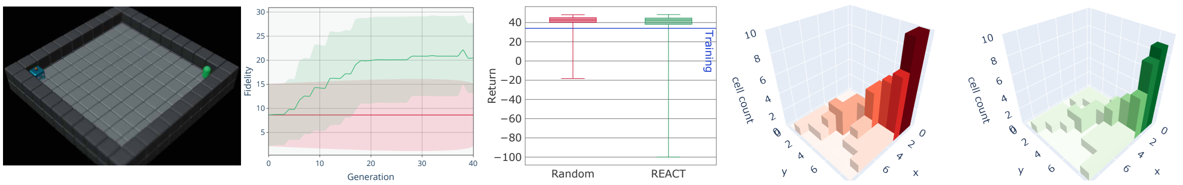
Setup. To validate the proposed architecture, we use a simple, fully observable discrete *FlatGrid11* environment with 11×11 fields shown in Fig. 3a(Altmann, 2023). The goal of the policy is to reach the target state (rewarded $+50$), where there are neither holes nor obstacles that could disrupt the agent's path. A step cost of -1 is applied to encourage choosing the shortest path. Episodes are terminated upon reaching the target state or after 100 steps. We use PPO (Schulman et al., 2017) with default parameters (Raffin et al., 2021) to train a policy that we can then evaluate with REACT. To show diverse behavior, we intentionally terminated training early (after 35k steps), just after the agent confidently reached the target. Using such an imperfect policy has a higher probability that the agent has not yet explored the entire environment. For evaluation purposes, we also want it to display behavior that leads the agent not to reach its goal. Note that the policy is trained with a single initial state shown in Fig. 3a. The following results are averaged over ten random seeds to increase the significance of the experimental results presented to optimize the demonstrations based on a single, previously trained policy.

Metrics. To provide an intuition over the resulting demonstrations $\mathcal{T}$, we summarize them in a single *3D histogram*, displaying the state-frequency of all grid cells. Compared to showing the discrete paths (cf. Fig. 1), this allows the visualization of results over multiple optimization runs without diminishing the depiction of the demonstration diversity by averaging them. Since viewing the behavior diversity of the final demonstrations is very subjective, we additionally consider the cumulated demonstration fidelity:

$$S = \sum_{\tau \in \mathcal{T}} \frac{|\tau|}{|\mathcal{T}|} \left|\bar{R} - r_\tau\right|, \tag{8}$$

with the absolute mean reward $\bar{R} = \frac{1}{|\mathcal{T}|} \sum |r_\tau|$ and the total trajectory length $|\mathcal{T}| = \sum |\tau|$, adapted from (Guo et al., 2021). Intuitively, the fidelity of an explanation measures the approximation quality w.r.t the given model, where a higher value indicates higher coverage (Molnar, 2020). As we consider a set of trajectories to serve as an explanation, S could also be viewed as their Shapley values, i.e., the impact of each trajectory on the total demonstration (Shapley and Shubik, 1954). Consequently, S also closely resembles the population diversity $\mathcal{D}$, defined earlier. Furthermore, we consider the *final return* and *final (trajectory) length*. Both metrics are crucial when training the optimal policy (maximizing the return while minimizing the solution length) and do not influence the optimized fitness function. However, we are not interested in the minimum or maximum of the returns or lengths but instead in the range of the metrics and how uniformly the individuals are spread across different returns and trajectory lengths. Therefore, we use box plots to visualize our results, where a bigger range between the whiskers promises greater diversity, and larger boxes indicate an even distribution. We also report the deterministic policy performance in the unaltered training environment, which is often used to validate learned behavior and serves as a baseline. Note that the fidelity for a single trajectory with any contrasting behavior is always zero. This already accurately reflects the deficiencies of considering a single training scenario for the evaluation. Furthermore, we compare REACT to a random search approach, implemented as the initial population $\mathbb{P}_0$ before applying the evolutionary process. This *Random* approach could be considered most closely related to comparable interpretability approaches, altering the environment without optimization while maintaining comparability to REACT.

Results. Fig. 3 shows the evaluation results. The trained policy reaches a return of 34 with a trajectory length of 16 (cf. Fig. 3c). Using a random pool of



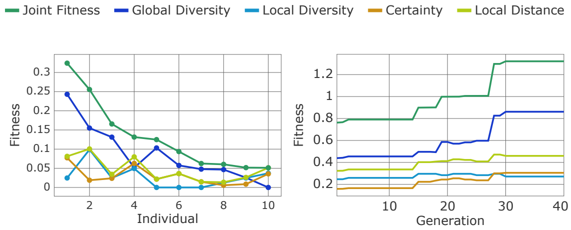
(a) FlatGrid11 (b) Fidelity (c) Final Return (d) Random $\mathcal{T}$ (e) REACT $\mathcal{T}$

Figure 3: REACT Evaluation: Fidelity (c) and Final Return (c) of Random (d) and REACT (e) demonstrations of a PPO policy trained for 35k steps in the FlatGrid11 (a) show increased diversity and even distribution of REACT-generated demonstrations over random or static initial states, with the training performance in the unaltered environment shown as a blue line in (c).

initial states increases the encountered return while reducing the trajectory length of the resulting demonstrations by moving the initial state closer to the target. Yet, random demonstrations still mostly yield behavior in the upper reward region. REACT manages to diversify the pool of demonstrations further, more evenly covering a larger region of final returns. Looking at the final fidelities in Fig. 3b, REACT is able to double the demonstration quality compared to the Random approach. Analyzing the resulting demonstrations from a single population, shown in Figs. 3d and (e), reveals two further insights: Overall, most trajectories successfully reach the target, shown by the highest occurrence of the target state, indicating a successfully trained policy that is robust to the introduced state disturbances. Yet, REACT produces more diverse trajectories distributed over farther states. Some states even resulted in the policy failing to navigate to the target, as indicated by outliers with a final return of -100.

Fitness Impact. Besides yielding diverse demonstrations, we also want to ensure the appropriateness of the proposed *joint fitness*. Fig. 4 therefore provides an additional in-depth analysis of the impact of the fitness components across the single last population of 10 individuals (a) and throughout the 40 optimization generations (b).

To accurately show the influence of the *local diversity* (light blue) and the *certainty* (orange), we visualize their population distance, which is combined in the minimum *local distance* (yellow) to be accumulated with the *global diversity* (blue) (cf. Eq. (6)). Interestingly, already with a population size of 10, individual fitness decreases throughout the population, reaffirming the chosen population size. Individuals evaluated with a lower global fitness (baring higher similarity to the overall population) show higher local distances, i.e., dissimilarities to the population regarding the diversity of the behavior itself, which conceptually justifies considering both diversity perspectives. In addition, all fitness components are shown to influence the whole behavior optimization, evenly increasing throughout the 40 generations. The considerably minor improvement in the last ten generations indicates convergence of the optimized demonstrations.



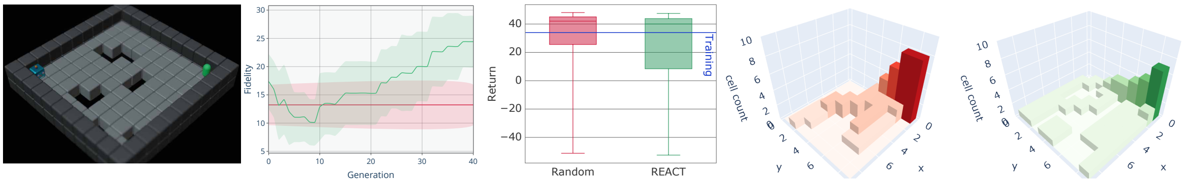
(a) Population Analysis (b) Generation Analysis

Figure 4: FlatGrid *JointFitness* Analysis.

Holey Gridworld

To further evaluate our approach, we use the more complex *HoleyGrid* environment shown in Fig. 5a, extending the previous *FlatGrid* with holes immediately terminating an episode with a reward of -50. The holes add additional complexity to the gridworld since the policy needs to learn to circumvent them to reach the target successfully. The policy to be analyzed is trained with PPO for 150k steps in a static layout, just reaching successful behavior, with a return of 36 and a trajectory length of 14 (cf. Fig. 5c).

The evaluation results in Fig. 5 reveal a smaller range of returns than the FlatGrid results, presumably caused by the additional holes. In contrast to the unaltered training environment in which the policy navigates successfully, we are able to reveal unsuccessful behavior with returns slightly below -50. Again, REACT covers a slightly larger fraction of the return compared to demonstrations from randomly generated initial states. Regarding their fidelity (cf., Fig. 5b), the final REACT demonstrations significantly outperform the Random demonstrations with a mean of around 24, even though dropping slightly below the Random baseline at around 13 in the initial generations. This is also reflected in the demonstration 3D histograms in Figs. 5d and (e). REACT



(a) HoleyGrid11 (b) Fidelity (c) Final Return (d) Random $\mathcal{T}$ (e) REACT $\mathcal{T}$

Figure 5: HoleyGrid Evaluation: Fidelity (b) and Final Return (c) of Random (d) and REACT (e) demonstrations of a PPO policy trained for 150k steps in the HoleyGrid11 (a) indicate further edge-case demonstration being generated using REACT over random or static initial states. The blue line in (c) displays the training performance in the unaltered environment.

demonstrations almost cover the whole state space, where, due to the nature of the fitness, we can assume all remaining states to yield comparable behavior that would not increase the demonstration diversity. The unoptimized demonstrations only cover more direct solution paths, which is also reflected in the smaller interquartile range of the according returns. Please refer to the appendix for an in-depth analysis of optimization progress and the impact of joint fitness.

Combining the high demonstration coverage of 10 highly diverse yet comprehensibly compact trajectories, we argue that REACT allows a human to properly assess the trained policy's inherent behavior. Concretely, the analyzed policy can be described as robust with high certainty, given the above-zero interquartile range of the demonstration returns, where further training in some problematic edge cases could be desirable depending on the intended application. Overall, REACT increases the interpretability of the policy at hand, especially compared to a single training trajectory with randomly chosen initial positions.

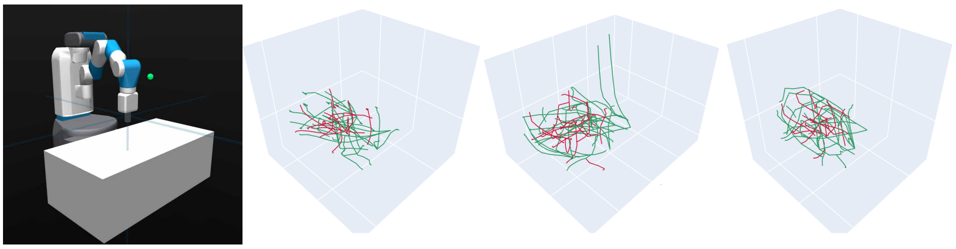
Continous Robotic Control

Finally, we demonstrate the effect of REACT in a more complex real-world application, where it could be utilized to decide between deploying different policies. For this, we use the continuous robotic control environment *FetchReach* shown in Fig. 6a.

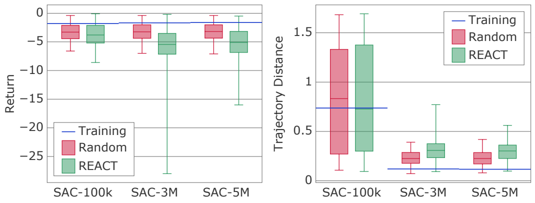
Environment. The agent is represented by a manipulator, the robotic arm, with six degrees of freedom, and its end effector, a gripper. The task is to control the robotic arm by applying a three-dimensional force vector to move the gripper to reach the target state (green point). In contrast to the previous gridworld environments, both action- and observation-space are real-valued. Furthermore, the task is open-ended such that episodes continue for 50 steps regardless of successfully reaching the target. Therefore, we use a sparse reward function, where the agent is penalized -1 for every step in which it is not close to the target, i.e., where the Euclidean distance between the effector and the target is greater than 0.05. During training, the effector's position is always initialized at the center, while the target is randomly positioned within a 0.3-sized cube around the center to improve generalization of the learned behavior (de Lazcano et al., 2023).

REACT Parameters. Given the increased environmental complexity, we adapted the parameterization of REACT according to preliminary studies. Most importantly, to remove all random factors from generating demonstrations, we include the target position and the agent (gripper) position in the initial state to be optimized. This results in a 6-dimensional state encoding, which we encode with a bit-length of 9 to reduce the intervals between possible states to less than 0.001. Furthermore, we replace the total number of possible states $|\{s \in \mathcal{S}\}|$ for calculating the local diversity (cf. Eq. (1)) by the trajectory length $|\tau|$, which results in the static horizon $H = 50$ for this environment. Also, as previously denoted, we use a discretization of states $s \in \tau$ such that the visited fraction of the state space remains reflected as intended. Finally, we increased the population size to 30 and used 1000 generations. Lastly, instead of analyzing a single moderately trained policy, we compare policies from three stages of training.

Training. Having proven beneficial in various continuous control tasks, we train the policies to be compared using *SAC* (Haarnoja et al., 2018), implemented with default parameterization (Raffin et al., 2021). To demonstrate the comparative evaluation capabilities, we trained policies for 100k, 3M, and 5M steps, which we refer to as *SAC-100k*, *SAC-3M*, and *SAC-5M* respectively for the following. Therefore, we are able to compare policies from three training stages, ranging from early convergence to possibly over-trained, thus overfitting the training task.



(a) FetchReach (b) SAC-100k $\mathcal{T}$ (c) SAC-3M $\mathcal{T}$ (d) SAC-5m $\mathcal{T}$



(e) Final Return (f) Final Length

Env	Algorithm	Random	REACT
FlatGrid11	PPO-35k	8.62 ± 6.47	$\mathbf{20.48 \pm 7.25}$
HoleyGrid11	PPO-150k	13.22 ± 3.58	$\mathbf{24.40 \pm 4.57}$
FetchReach	SAC-100k	1.44 ± 0.15	$\mathbf{1.61 \pm 0.23}$
	SAC-3M	1.46 ± 0.22	$\mathbf{7.39 \pm 4.92}$
	SAC-5M	1.46 ± 0.24	$\mathbf{2.83 \pm 0.93}$

(g) Mean final fidelities and 95% confidence intervals

Figure 6: REACT Evaluation: Final Return (e) and Length (f) of Random (Red) and REACT (Green) demonstrations of SAC policies trained in the FetchReach (a) environment for 100k (b), 3M (c), and 5M (d) steps demonstrate the applicability of REACT in discerning policies from different training stages by disclosing their inherent behavior. (g) summarizes our results.

Results. The overall evaluation results are shown in Fig. 6. The performance of all models in the unaltered training environment shows both increasing returns of around -1.8, -1.7, and -1.6 and decreasing trajectory distances of around 0.73, 0.12, and 0.11 for SAC-100k, SAC-3M, and SAC-5M respectively. With a maximum target distance of 0.3, based only on these results, the primal SAC-100k could be disregarded due to the significantly more extensive movement, even though reaching competitive rewards. However, REACT reveals further important insights on which to base model interpretations and subsequent decisions. Regarding the final trajectory length, REACT shows diverse demonstrations to be evenly distributed around the single training experience for SAC-100k, with both the length and the variance of the length decreasing upon further training. Compared to demonstrations based on random initial states, REACT again shows a slight increase in the overall diversity and even distribution of demonstrations. More interesting results, however, are shown for the final return, where random configurations, similar to the training configuration, do not reveal any insightful differences between the models. On the other hand, REACT reveals the overall return variance increasing with further training, with the median of returns even decreasing. It is important to note that the return is not included in the fitness to optimize the demonstrations. Thus, these observations emerge from diverse behavior generated by the policies. Given the increasing returns observed for the training configuration, this could most likely be explained as overfitting behavior. To give an intuition of the scope and nature of the resulting demonstrations, we finally consider the path of all Random (red) and REACT (green) trajectories shown in Figs. 6b-(d). Due to the continuous nature of the environment and the increased number of individuals, we did not plot the resulting demonstrations as cumulative distributions (mainly because averaging would diminish any diversity within the populations). Although this kind of visualization does not allow for the precise analysis of each resulting trajectory, it perfectly conveys the overall nature of the generated demonstrations[2]. Again, REACT covers a comparably larger fraction of the state space more evenly and even detected a policy insufficiency of SAC-3M, causing the demonstration of an outlier. In summary, the shortest-trained policy reaches targets the fastest, showing the lowest penalties and thus the highest returns but with the lowest precision and, hence, the highest movement and trajectory length. Longer-trained policies, on the other hand, show higher penalties. They reach the target slower yet more precisely, as indicated by the overall lower trajectory length. The assessment of those characteristics heavily depends on the intended application; however, REACT has revealed those critical characteristics of the inherently learned behavior.

Finally, Table 6g summarizes the final fidelities. Overall, REACT improves the demonstration quality compared to the Random baseline, roughly maintaining its low score throughout the different models.

[2] Video renderings are available at https://github.com/philippaltmann/REACT.

Notably, REACT extracts viable characteristics, especially for more mature models, significantly outperforming the chosen baseline and showcasing its scalability.

7 CONCLUSION

To enhance the interpretability of RL, we introduced *Revealing Evolutionary Action Consequence Trajectories* (REACT). REACT adds disturbances to the environment by altering the initial state, causing the policy to generate edge-case demonstrations. To assess trajectories for demonstrating a given policy, we formalized a joint fitness combining the local diversity and certainty of the trajectory itself with the global diversity of a population of demonstrations. To optimize a pool of demonstrations, we apply an evolutionary process to the population of individuals, encoded as the initial state, evaluated by the joint fitness. To evaluate REACT, we analyzed various policies trained in flat and holey gridworlds as well as a continuous robotic control task at different training states. Comparisons to the unaltered training environment and randomly generated initial states showed that REACT reveals a set of more diverse and more evenly distributed demonstrations to serve as a varietal basis to assess the learned (inherent) behavior. In addition to the final return, we analyzed the demonstrations' utility using an adapted fidelity metric. However, we refrain from human evaluations and leave the subjective assessment of the appended demonstrations to the reader. Furthermore, we only introduced disturbances of the initial agent and target positions. Thus, future work should examine extending REACT to further variations of the environment, such as the overall layout or the task itself. Also, the resulting pool of demonstrations could be used either to further improve the policy regarding revealed vulnerabilities or to infer a global causality model to further foster the policy's interpretability. Overall, we believe that REACT represents a universal policy-centric starting point for improving the overall interpretability of the currently mostly opaque RL models.

ACKNOWLEDGEMENTS

This work was partially funded by the Bavarian Ministry for Economic Affairs, Regional Development and Energy as part of a project to support the thematic development of the Institute for Cognitive Systems.

REFERENCES

Alharin, A., Doan, T.-N., and Sartipi, M. (2020). Reinforcement learning interpretation methods: A survey. *IEEE Access*, 8:171058–171077.

Altmann, P. (2023). hyphi gym. https://github.com/philippaltmann/hyphi-gym/.

Altmann, P., Ritz, F., Feuchtinger, L., Nüßlein, J., Linnhoff-Popien, C., and Phan, T. (2023). Crop: towards distributional-shift robust reinforcement learning using compact reshaped observation processing. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23.

Amir, D. and Amir, O. (2018). Highlights: Summarizing agent behavior to people. In *Adaptive Agents and Multi-Agent Systems*.

Behrens, M., Gube, M., Chaabene, H., Prieske, O., Zenon, A., Broscheid, K.-C., Schega, L., Husmann, F., and Weippert, M. (2023). Fatigue and human performance: an updated framework. *Sports medicine*, 53(1):7–31.

Bhatt, V., Tjanaka, B., Fontaine, M., and Nikolaidis, S. (2022). Deep surrogate assisted generation of environments. *Advances in Neural Information Processing Systems*, 35:37762–37777.

Cobbe, K., Hesse, C., Hilton, J., and Schulman, J. (2020). Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR.

de Lazcano, R., Andreas, K., Tai, J. J., Lee, S. R., and Terry, J. (2023). Gymnasium robotics. http://github.com/Farama-Foundation/Gymnasium-Robotics.

Fogel, D. B. (2006). *Evolutionary computation: toward a new philosophy of machine intelligence*. John Wiley & Sons.

Gabor, T. and Altmann, P. (2019). Benchmarking surrogate-assisted genetic recommender systems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19, page 1568–1575, New York, NY, USA. Association for Computing Machinery.

Gabor, T., Belzner, L., and Linnhoff-Popien, C. (2018). Inheritance-based diversity measures for explicit convergence control in evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 841–848.

Gabor, T., Sedlmeier, A., Kiermeier, M., Phan, T., Henrich, M., Pichlmair, M., Kempter, B., Klein, C., Sauer, H., AG, R. S., et al. (2019). Scenario co-evolution for reinforcement learning on a grid world smart factory domain. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 898–906.

Guo, W., Wu, X., Khan, U., and Xing, X. (2021). Edge: Explaining deep reinforcement learning policies. *Advances in Neural Information Processing Systems*, 34:12222–12236.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290.

Heuillet, A., Couthouis, F., and Díaz-Rodríguez, N. (2021). Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214:106685.

Huang, S. H., Bhatia, K., Abbeel, P., and Dragan, A. D. (2018). Establishing appropriate trust via critical states. *CoRR*, abs/1810.08174.

Huang, S. H., Held, D., Abbeel, P., and Dragan, A. D. (2017). Enabling robots to communicate their objectives. *CoRR*, abs/1702.03465.

Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, pages 2419–2426. IEEE.

Khadka, S. and Tumer, K. (2018). Evolutionary reinforcement learning. *CoRR*, abs/1805.07917.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Lage, I., Lifschitz, D., Doshi-Velez, F., and Amir, O. (2019). Exploring computational user models for agent policy summarization. *CoRR*, abs/1905.13271.

Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.

Li, X., Xiong, H., Li, X., Wu, X., Zhang, X., Liu, J., Bian, J., and Dou, D. (2022). Interpretable deep learning: interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*.

Lin, B. and Su, J. (2008). One way distance: For shape based similarity search of moving object trajectories. *GeoInformatica*, 12:117–142.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Miller, B. L., Goldberg, D. E., et al. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212.

Molnar, C. (2020). *Interpretable machine learning*. Lulu.com.

Neumann, A., Gao, W., Wagner, M., and Neumann, F. (2019). Evolutionary diversity optimization using multi-objective indicators. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 837–845.

Pang, Q., Yuan, Y., and Wang, S. (2022). Mdpfuzz: testing models solving markov decision processes. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 378–390.

Parker-Holder, J., Jiang, M., Dennis, M., Samvelyan, M., Foerster, J., Grefenstette, E., and Rocktäschel, T. (2022). Evolving curricula with regret-based environment design. In *International Conference on Machine Learning*, pages 17473–17498. PMLR.

Parker-Holder, J., Pacchiano, A., Choromanski, K., and Roberts, S. (2020). Effective diversity in population-based reinforcement learning. *CoRR*, abs/2002.00632.

Pleiss, G., Zhang, T., Elenberg, E., and Weinberger, K. Q. (2020). Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056.

Puterman, M. L. (1990). Markov decision processes. *Handbooks in operations research and management science*, 2:331–434.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Richard S. Sutton, A. G. B. (2014, 2015). *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts, London, England, 2 edition.

Rolf, B., Jackson, I., Müller, M., Lang, S., Reggelin, T., and Ivanov, D. (2023). A review on reinforcement learning algorithms and applications in supply chain management. *International Journal of Production Research*, 61(20):7151–7179.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

Sequeira, P. and Gervasio, M. (2020). Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations. *Artificial Intelligence*, 288:103367.

Shapley, L. S. and Shubik, M. (1954). A method for evaluating the distribution of power in a committee system. *American political science review*, 48(3):787–792.

Tappler, M., Córdoba, F. C., Aichernig, B. K., and Könighofer, B. (2022). Search-based testing of reinforcement learning. *arXiv preprint arXiv:2205.04887*.

Vartiainen, P. (2002). On the principles of comparative evaluation. *Evaluation*, 8(3):359–371.

Wineberg, M. and Oppacher, F. (2003). The underlying similarity of diversity measures used in evolutionary computation. In *Genetic and evolutionary computation conference*, pages 1493–1504. Springer.

Wu, S., Yao, J., Fu, H., Tian, Y., Qian, C., Yang, Y., FU, Q., and Wei, Y. (2023). Quality-similar diversity via population based reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

Wurman, P. R., Barrett, S., Kawamoto, K., MacGlashan, J., Subramanian, K., Walsh, T. J., Capobianco, R., Devlic, A., Eckert, F., Fuchs, F., et al. (2022). Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228.

Zolfagharian, A., Abdellatif, M., Briand, L. C., Bagherzadeh, M., and Ramesh, S. (2023). A search-based testing approach for deep reinforcement learning agents. *IEEE Transactions on Software Engineering*, 49(7):3715–3735.